

Technology Report (TR)

Intelligent Lifestyle

Team Daedalus (`s440gf`)

Dominic Xavier Mendonca - (`dxm`)
Glenn Maxwell Fry - (`gmfr`)
Ho-Jung Simon Youn - (`hjsy`)
Masyuri Jaya Tjhandana - (`masyurit`)
Mei Ling Leong - (`mlleong`)
Quyên Lê Quach - (`qlq`)
Saw Ai Soon - (`sasoon`)
Yat Ivan Wong - (`ywong`)

Revision: 1.0.0
25 October 2004

Maintained by: Ho-Jung Simon Youn - (`hjsy`)

Abstract

This document details the technologies which could be employed in implementations of agent systems. It includes descriptions of past uses of each technologies, the use Team Daedalus put the technology to and other possible uses of the technology.

Contents

- 1 Introduction 7**
 - 1.1 Purpose 7
 - 1.2 Scope 7
 - 1.3 Intended Audience 7
 - 1.4 Project Overview 7
 - 1.5 Personnel 8
 - 1.5.1 Development Team 8
 - 1.5.2 Supervisor 8
 - 1.5.3 Clients 8
 - 1.6 Definitions and Acronyms 8
 - 1.6.1 Definitions 9
 - 1.6.2 Acronyms 9
 - 1.7 Reference Documents 9
 - 1.8 References 10

- 2 May 7th Demonstration 11**
 - 2.1 Bluetooth: D-Link 11
 - 2.1.1 Overview 11
 - 2.1.2 Current uses of this technology 11
 - 2.1.3 What we used this technology for 11
 - 2.1.4 How successful were we? 11
 - 2.1.5 Other possible uses of this technology 12
 - 2.2 Imaging 13
 - 2.2.1 Overview 13
 - 2.2.2 Current uses of this technology 13
 - 2.2.3 What we used this technology for 13
 - 2.2.3.1 Face Detection 13
 - 2.2.3.2 Face Recognition 14
 - 2.2.4 How successful were we? 14
 - 2.2.5 Other possible uses of this technology 14
 - 2.3 PDA 15
 - 2.3.1 Overview 15
 - 2.3.2 Current uses of this technology 15
 - 2.3.3 What we used this technology for 15
 - 2.3.4 How successful were we? 15
 - 2.3.5 Other possible uses of this technology 15
 - 2.4 Speech Recognition 17
 - 2.4.1 Overview 17
 - 2.4.2 Current uses of this technology 17
 - 2.4.3 What we used this technology for 17
 - 2.4.4 How successful were we? 17
 - 2.4.5 Other possible uses of this technology 17
 - 2.5 Text-to-Speech 18
 - 2.5.1 Overview 18
 - 2.5.2 Current uses of this technology 18
 - 2.5.3 What we used this technology for 18

2.5.4	How successful were we?	18
2.5.5	Other possible uses of this technology	18
3	Final Product	19
3.1	Bluetooth Adaptor: Clarinox	19
3.1.1	Overview	19
3.1.2	Current uses of this technology	19
3.1.3	What we used this technology for	19
3.1.4	How successful were we?	19
3.1.5	Other possible uses of this technology	20
3.2	Imaging	21
3.2.1	Overview	21
3.2.2	Current uses of this technology	21
3.2.3	What we used this technology for	21
3.2.4	How successful were we?	21
3.2.5	Other possible uses of this technology	22
3.3	JADE	23
3.3.1	Overview	23
3.3.2	Current uses of this technology	23
3.3.3	What we used this technology for	24
3.3.4	How successful were we?	24
3.3.5	Other possible uses of this technology	24
3.4	PDA	25
3.4.1	Overview	25
3.4.2	Current uses of this technology	25
3.4.3	What we used this technology for	25
3.4.4	How successful were we?	25
3.4.5	Other possible uses of this technology	25
3.5	Recording	26
3.5.1	Overview	26
3.5.2	Current uses of this technology	26
3.5.3	What we used this technology for	26
3.5.4	How successful were we?	26
3.5.5	Other possible uses of this technology	26
3.6	Speech-to-Text	27
3.6.1	Overview	27
3.6.2	Current uses of this technology	27
3.6.3	What we used this technology for	27
3.6.4	How successful were we?	27
3.6.5	Other possible uses of this technology	27
3.7	Text-to-Speech	29
3.7.1	Overview	29
3.7.2	Current uses of this technology	29
3.7.3	What we used this technology for	29
3.7.4	How successful were we?	29
3.7.5	Other possible uses of this technology	29
3.8	Web server	30
3.8.1	Overview	30

3.8.2	Current uses of this technology	30
3.8.3	What we used of this technology	30
3.8.4	How successful were we	30
3.8.5	Other possible uses of this technology	30
4	Change Log	31

List of Figures

List of Tables

1	Development Team	8
2	Change Log	31

1 Introduction

The Technology Report (TR) documents the technologies that could be employed in implementations of agent systems, especially in regard to an intelligent lifestyle. It includes descriptions of past uses of each technologies, the use Team Daedalus put the technology to and other possible uses of the technology.

1.1 Purpose

The purpose of the TR is to provide a formal documentation of technologies used for the purpose of demonstrating our system for both the May 7th presentation and the final product. The TR may serve as a reference to future projects that extend on the capabilities of the Intelligent Lifestyle project.

1.2 Scope

The scope of the TR is to document all technologies used by Team Daedalus for the purpose of demonstrating our system for both the May 7th presentation and the final product.

1.3 Intended Audience

The intended audience for the Technology Report are members of Team Daedalus, the Supervisor, and the Client.

1.4 Project Overview

The aim of the Intelligent Lifestyle project is

- **To design and build a system via the ROADMAP methodology comprising of some intelligent agents, for the explicit purpose of providing demonstrations of Intelligent Agents.**

This aim is an attempt to balance the two requirements from the Clients. This is necessary as meeting both fully would be impossible with current time constraints. The individual aims of the Clients are shown below.

1. To provide a demonstration of intelligent agents.

The Clients wish to have something that can demonstrate agents, agent behavior and intelligence. Scenarios will be used to demonstrate these features.

2. To implement the ROADMAP methodology(section [12.3.2](#)) and create an intelligent agent system.

This Clients wish to test out ROADMAP and produce an example of an intelligent agent system. They believe this methodology will be useful in implementing the project.

1.5 Personnel

1.5.1 Development Team

The development team of this project is Team Daedalus, enrolled in the subject of the Advanced Software Engineering Projects.

Name	Login ¹	Phone no.
Carol Poon	cyspoon	0401-959-660
Dominic Mendonca	dxm	0411-093-253
Glenn Fry	gmfr	0418-372-176
Simon Youn	hjsy	0403-438-830
Jian Alan Huang	jhua	0402-001-910
Kieran Simpson	kieranjs	0412-821-128
Masyuri Tjhandana	masyurit	0413-150-311
Mei Ling Leong	mllleong	0413-689-314
Nathaporn Eiamvittayakorn	neiam	0407-565-824
Quyen Quach	qlq	0412-122-031
Shirley Soon	sasoon	0407-552-338
Wendy Tsang	wwttsang	0411-199-822
Ivan Wong	ywong	0411-863-261

Table 1: Development Team

1.5.2 Supervisor

Kendall Lister
Department of Computer Science and Software Engineering
University of Melbourne
krl@cs.mu.oz.au

1.5.3 Clients

Leon Sterling
Department of Computer Science and Software Engineering
University of Melbourne
leon@cs.mu.oz.au

Thomas Juan
Department of Computer Science and Software Engineering
University of Melbourne
tlj@cs.mu.oz.au

1.6 Definitions and Acronyms

This section describes all definitions and acronyms used by Team Daedalus in this document.

¹Email addresses of team members can be derived from the user's login name by appending @students.cs.mu.oz.au.

1.6.1 Definitions

Agent

Pieces of code in an system that can detect changes in its environment.

Bluetooth

A short-range radio technology for Internet and mobile devices.

OpenCV library

Open Source Computer Vision library created by Intel.

WLAN

Wireless local area network technology providing short-range, high-speed data connections between mobile data devices and nearby WLAN access points.

1.6.2 Acronyms

<i>BDI</i>	Belief Desire Intention
<i>FIPA</i>	Foundation for Intelligent Physical Agents
<i>GPS</i>	Global Positioning System
<i>HTTP</i>	Hyper Text Transfer Protocol
<i>IRDA</i>	Infrared Data Association
<i>IS</i>	Information System
<i>JADE</i>	Java Agent Development Framework
<i>JNI</i>	Java Native Interface
<i>PAN</i>	Personal Area Network
<i>PC</i>	Personal Computer
<i>PDA</i>	Personal Digital Assistant
<i>POP</i>	Point Of Presence
<i>ROADMAP</i>	Role Oriented Analysis and Design for Multi-Agent Programming
<i>SADD</i>	Software Architecture Design Document
<i>SDD</i>	Software Design Document
<i>SDK</i>	Software Development Kit
<i>SPMP</i>	Software Project Management Plan
<i>SRS</i>	Software Requirements Specifications
<i>STT</i>	Speech To Text
<i>TTS</i>	Text To Speech
<i>USB</i>	Universal Serial Bus
<i>VB</i>	Visual Basic
<i>WLAN</i>	Wireless Local Area Network
<i>XML</i>	eXtensible Markup Language

1.7 Reference Documents

The following documents should be read in parallel to assist the understanding of this document.

1. Team Daedalus Software Architecture Design Document (SADD)
2. Team Daedalus Software Design Document (SDD)

3. Team Daedalus Software Requirements Specification (SRS)

1.8 References

This section specifies the external material such as standards, textbooks and other documents that have been referenced to assist the creation of this document.

1. *Bluetooth Overview*,
<http://www.thewirelessdirectory.com/Bluetooth-Overview/Bluetooth-Overview.htm>
2. *Clarinox*,
<http://www.clarinox.com.au/>
3. *Clove Technology - HP iPAQ 5550*,
http://www.clove.co.uk/products/products.asp?strAreaNo=400_1_2&intElement=1936
4. *Intel Research*,
<http://www.intel.com/research/mrl/research/opencv/overview.htm>
5. *JADE*,
<http://jade.tilab.com/>
6. *Java Media Framework API*,
<http://java.sun.com/products/java-media/jmf/index.jsp>
7. *Speech SDK 5.1 for Windows applications*,
<http://www.microsoft.com/speech/download/sdk51/>
8. *Sphinx OCX*,
<http://www.speech.cs.cmu.edu/inner/sphinxocx/SphinxOCX.html>

2 May 7th Demonstration

This section describes technologies that were used by Team Daedalus for the purpose of our May 7th Demonstration.

2.1 Bluetooth: D-Link

2.1.1 Overview

Bluetooth is a wireless technology that enables a device to wirelessly communicate in the 2.5 GHz frequency band. It allows devices such as mobile phones, headsets, PDAs and computers to communicate and send data to each other without the need for wires or cables to link devices together. It has been specifically designed as a low cost, low power, radio technology, which is particularly suited to the short range Personal Area Network (PAN) application.

The bluetooth adaptor from D-Link, model DBT-120 was selected for the purpose of our project due to the unavailability of the Belkin adaptor which was our first choice. However it was decided that the D-Link adaptor would make an appropriate substitute due to its features which were similar to the Belkin adaptor.

2.1.2 Current uses of this technology

The primary uses of bluetooth technology is currently as follows:

1. wireless cable replacement for devices such as keyboards, mouses
2. voice and data access points for using devices such as wireless headsets with a mobile phone, transferring data between devices such as mobile phones and PDAs, between PDAs, or between a desktop computer and a PDA
3. Ad-hoc or temporary networking between bluetooth equipped devices

2.1.3 What we used this technology for

The D-Link bluetooth adaptor was used to transfer an image from a laptop computer to a PDA via a wireless bluetooth connection. More specifically, bluetooth technology was used to demonstrate an intruder scenario, where if an unidentified person was registered by our imaging program (see section 2.2) the person's image would be transmitted wirelessly using bluetooth to the home owner's PDA. This was made possible by using our D-Link adaptor connected to the USB port of a laptop computer in conjunction with software that was developed by Team Daedalus to complete the transmission. On the PDA, bluetooth connectivity was also enabled, and again software developed by team Daedalus was used to receive the transmission.

2.1.4 How successful were we?

The D-Link adaptor worked flawlessly in pairing up with the iPaq PDA and transmitting the required image to the PDA. Difficulties were however encountered in developing low-level functionalities such as detecting the presence of all bluetooth devices in the vicinity since the drivers supplied with the D-Link adaptor did not fully support the Windows XP bluetooth stack. We determined that to produce more sophisticated functionalities with bluetooth, if we wished to continue to use Visual Studio .Net 2003 we would need to purchase a bluetooth adaptor that fully

supported the Windows XP bluetooth stack, or we would need to find an alternative bluetooth development kit altogether (see Clarinox bluetooth section [3.1](#)).

2.1.5 Other possible uses of this technology

No other possible uses of this technology has been identified.

2.2 Imaging

This section describes the application used to handle imaging related functionalities such as the ability to recognize the identity of a person captured using a USB camera. The application used for our purposes, OpenCV, was modified to suit our needs in this project.

2.2.1 Overview

OpenCV is a research topic that has been conducted by Intel. Due to the continual development of OpenCV, it is not yet a commercial product and is made available to developers as open source software. Since OpenCV is not commercial product it has not undergone the usual rigorous testing that is expected of software produced by a large corporation which means that it has instabilities that makes the software prone to crashing.

OpenCV is not a single program that does one specific function. Instead, OpenCV has several functions, all of which are related to imaging technologies. OpenCV is written in C++, making it a suitable for our team since it is a language that we have had experience in.

2.2.2 Current uses of this technology

OpenCV is used to detect faces and recognize them using a video stream such as a USB camera. This process can be automated and can be made to run continuously.

2.2.3 What we used this technology for

We used OpenCV to be able to perform:

1. face detection
2. face recognition

2.2.3.1 Face Detection

The face detection program can be made to run continuously while actively trying to detect faces in the environment. It does not detect or track motion. Instead, the program tries to look for faces in the environment and indicates the detection with a rectangle drawn around the face. If the face is being detected continuously, it would seem that the program actually tracks the face, however this is not the case. What the program is in fact doing is continuously detecting faces at a fast rate.

With the use of a USB camera, face detection wasn't found to be as reliable in detecting the presence of a face due to its limited resolution. The way OpenCV detects faces is by trying to obtain the eyes of the object, which means that the person needs to look directly at the camera so it can be detected. This means that the person can't be too far from the camera, and is effectively useless in detecting faces at a distance greater than 2 meters for a VGA resolution USB camera.

In regards to our project, we modified the face detection code to save the image of the face captured to a file, which will become an input to the next function (recognition).

2.2.3.2 Face Recognition

The other OpenCV function we used was face recognition. Face recognition works by comparing existing images of faces in a database with the image of a video source (USB camera) and returns the image from the database that has the most similarity between them. This means that the system will always return an answer. even if it doesn't recognize the image, it will still produce an output that has the most similarity factors between them.

In future use of this program, we will need to modify the face recognition program so that it takes the input and explicitly rejects it if it doesn't recognize the image, instead of producing the best comparison result. Also, when it doesn't recognize the image, it will try to store the image into the database automatically.

2.2.4 How successful were we?

We were able to successfully implement both face detection and recognition separately during prototyping. However, this was not sufficient for our final product, since our final requirements specifies that both face detection and recognition be automated in the one process.

For the May demonstration, we successfully demonstrated our working face detection code, which upon detecting a user's face, froze the video feed from the USB camera and drew a red rectangle around the detected face.

2.2.5 Other possible uses of this technology

No other possible uses of this technology has been identified.

2.3 PDA

This section describes the technologies used to demonstrate the capabilities of mobile communication. For our purposes, a HP iPaq PDA with bluetooth was chosen.

2.3.1 Overview

The Hewlett Packard iPaq 5550s are one of the most advanced PDAs on the market today. The iPaq is slim and light, ideal for mobile use. It comes with a full colour active TFT LCD display with built in WLAN and bluetooth capabilities.

2.3.2 Current uses of this technology

PDAs are used for a variety of different purposes, ranging from its originally designed use as an organizer to an Internet browser, or even as a portable gaming device. With the power and multimedia capabilities of current PDAs they are able to perform most functions that can be done on a desktop computer.

PDAs equipped with bluetooth are often used to transfer data to and from other PDAs or desktop computers. This way, the user is able to synchronize and share information. Bluetooth is also used to connect the PDA to devices such as mobile phones to gain the ability to connect to the Internet. In this case the mobile phone's GPRS Internet connect is used with the mobile phone acting as a modem. This allows for truly portable Internet usage.

2.3.3 What we used this technology for

For the May 7th demonstration the PDA was required to:

1. transfer an image from a laptop computer to the PDA via a wireless bluetooth connection;
2. display a welcoming message upon arriving at the CS or IS booths via a WLAN connection.

2.3.4 How successful were we?

The transfer of an image from the laptop computer to the PDA was completed without any difficulties. The image was transmitted using a D-Link bluetooth adaptor connected to a laptop (see section 2.1) and received by the PDA using its built-in bluetooth adaptor and displayed on its screen.

Unfortunately the messaging feature was not deployable in time for the demonstration. This was due to difficulties encountered configuring the Internet proxy settings at the venue, and also because of the instability of the wireless connection of the venue. Messaging did however work as expected in the ICT building where we had tested the system.

2.3.5 Other possible uses of this technology

The iPaq PDA may have the following possible uses:

1. Universal remote control using IRDA
2. Full navigational capabilities using a GPS adaptor

The iPaq using bluetooth may have the following possible uses:

1. Connectivity to a bluetooth wireless headset for speech recognition
2. Wireless control of a desktop PC

2.4 Speech Recognition

This section documents the technologies used in speech recognition. The package chosen by Team Daedalus is the CMU Sphinx Open Source Speech Recognition Tool.

2.4.1 Overview

Sphinx was developed by the Sphinx Group at Carnegie Mellon University. Sphinx-2 is a real-time, large vocabulary, speaker independent speech recognition system available under an Apache-style license.

2.4.2 Current uses of this technology

Mabel (the Mobile Table) is a robotic system that can perform waypoint and vision guided navigation, speech generation, speech recognition, natural language understanding, person finding, face finding and face following. Mabel was the winner of the 2002 robot host event at AAAI's 2002 conference at Edmunton, Canada.

Mabel used Sphinx to do speech recognition. Mabel could recognize about 400 sentences related to the food serving domain.

2.4.3 What we used this technology for

The Sphinx-2 OCX has been used by Team Daedalus to implement the speech recognition technology. This would allow for users of our system to verbally interact with it.

2.4.4 How successful were we?

We successfully implemented the speech recognition feature and the application was able to recognize speech in a quiet environment. However, we did not purchase a noise-cancelling microphone and thus, could not run this application in an environment with noise.

In our final product, the speech recognition subsystem should be as robust as possible - it should be able to recognize speech at a high level of accuracy regardless of the noise level in the environment.

Besides this, we need to build up a very large corpus for the final product. (A sentence corpus contains all possible words and the general sentence structure of the expected input.) This corpus will be fed into Sphinx's Knowledge Base Tool to build a custom dictionary and language model. The problem with this Knowledge Base Tool is that it only accepts files with a maximum of 5000 lines. We will need to spend some time coming up with the possible words/sentences that the system may need to recognize.

2.4.5 Other possible uses of this technology

No other possible uses of this technology has been identified.

2.5 Text-to-Speech

This section describes the technologies used to demonstrate text-to-speech.

2.5.1 Overview

Team Daedalus considered using Microsoft Speech SDK 5.1 to implement text-to-speech. However, the speech generated was found to be robot-like and impersonal. Therefore, we made a decision to record what we want to say in a WAV file instead.

2.5.2 Current uses of this technology

The Microsoft text-to-speech software can be used to 'read' information on the screen such as message boxes or any other form of text to the visually impaired. However, there are better (but expensive) software available for such purposes that have a more natural flowing speech, hence leaving the Microsoft text-to-speech software as more of a novelty product.

2.5.3 What we used this technology for

We used this technology when the system wanted to query the user or provide information to the user. Using text-to-speech software meant that the user wasn't required to have access to a screen hence allowing him/her to interact with the system in a more natural manner.

2.5.4 How successful were we?

The recordings were quite clear. However, it is strongly suggested that a proper text-to-speech package is used for the final product. We could use Microsoft Speech SDK 5.1, or we could simply use IBM ViaVoice. This way our system wouldn't be 'hard-coded'.

2.5.5 Other possible uses of this technology

No other possible uses of this technology has been identified.

3 Final Product

This section details technologies that were used by Team Daedalus for the development of our final product.

3.1 Bluetooth Adaptor: Clarinox

3.1.1 Overview

For an overview of bluetooth technology refer to section 2.1.

The Clarinox bluetooth adaptor was obtained due to the shortcomings of the D-Link bluetooth adaptor. Although both devices are similar in hardware, the Clarinox bluetooth adaptor development kit was found to be ideal for bluetooth software development as it comprises of a well documented API along with technical support and sample applications. This allowed Team Daedalus to more rapidly develop complex software for the purposes of our project, which could not be done using the D-Link adaptor and Microsoft Visual Studio .Net.

3.1.2 Current uses of this technology

See May 7th section detailing bluetooth technology, section 2.1.

3.1.3 What we used this technology for

For the purpose of our project, we used the Clarinox adaptor to:

1. detect the presence of bluetooth adaptors within its range (around 10 meters)
2. transfer a message containing text and an image to bluetooth devices within the Clarinox adaptor's range
3. act as a point of presence hub for guiding users within the ICT building

3.1.4 How successful were we?

The ability to detect the presence of bluetooth devices within the Clarinox bluetooth adaptor's range was supplied to us by Anton Kattan, an IS department software technician who had been working on an older serial port version of the Clarinox development kit from the start of 2004. The software he had supplied to us called POP (Point of Presence) worked flawlessly.

Sending text and image messages via the Clarinox adaptor (as opposed to the D-Link adaptor for the May 7th demonstration) required recoding the aforementioned POP software, which was relatively routine. We did however encounter an instability problem when multiple messages were sent to various PDAs continuously, which turned out to be a problem with the Clarinox bluetooth stack, since it was never designed for such use. The problem was never resolved, although having worked with the Clarinox technical leader was told that the bug would be fixed in the next version of the Clarinox development kit (version 1.2). On the receiving end, only minor modifications were needed to be made to the PDA software coded for the May 7th demonstration to get the PDA to receive both text and image messages.

Guiding comprised of using the Clarinox adaptor to send text only messages, which worked as expected. The same POP software developed for sending image and text messages were used for this purpose. Consequently, guiding was affected by the same issues of instability due to the Clarinox bluetooth stack.

3.1.5 Other possible uses of this technology

See May 7th section detailing bluetooth technology, section [2.1](#).

3.2 Imaging

This section describes the application used to handle imaging related functionalities. The application used for our purposes, OpenCV, was modified to suit our needs in this project.

3.2.1 Overview

See May 7th imaging, section [2.2](#).

3.2.2 Current uses of this technology

See May 7th imaging, section [2.2](#).

3.2.3 What we used this technology for

We used OpenCV to be able to perform:

1. face detection
2. face recognition

Further details on face detection and recognition can be found in section [2.2](#).

With the May 7th demonstration, both face detection and recognition were separate programs, requiring the user to manually input an image to the face recognition program when the user's image was captured by face detection. For the purpose of the final demonstration, we attempted to integrate these two functions to the one program which could be run continuously without any user intervention.

3.2.4 How successful were we?

A major problem encountered was recognizing the face detected. The way the face recognition works is by applying an algorithm to compare two images. This means that not only the face is being compared, but also the background. Therefore, the background of an image where face is detected, as well as the light intensity, was found to affect the recognition process. This could sometimes lead to 'wrong' output. However, we could minimize incorrect outputs by having a numerous images of persons recognized in the database. Also, the USB camera for detecting images would stay at the same place all the time. This means that the background and light intensity (assuming the USB camera is inside, unaffected by the weather, with continuous light source), would be the same all the time. This would then maximize the recognition process. We were unable to make these corrections to the software due to the limited time and resource availability.

In terms of integrating and automating face detection and recognition, we were unable to complete this task. Our main difficulties involved instability issues due to both programs using a significant amount of memory resulting in system crashes. For our final product, we instead automated the recognition of faces, where upon recognizing a face during a specified time period, the image of the person is sent to the Blackboard, where all JADE context information is posted.

3.2.5 Other possible uses of this technology

We could use another function of OpenCV, which is the 'motion tracking', to keep track of an object. Possibly, keeping a record on where the object has gone for the past length of times. This could be used for keeping records of intruder (on where he has gone in the house).

Currently face detection is unable to detect when the person looks away from the USB camera. We could then use the 'motion tracking' to keep detecting the face, even though the person doesn't look directly to the USB camera. A point may be put to the ears, or edges of the faces, and track its motion.

3.3 JADE

This section describes the technology used to set up communication between agents running independently on possibly different machines. We used the agent communication framework JADE for our implementation.

3.3.1 Overview

In order for individual autonomous agents to collaborate (an essential feature of a multi-agent system), a protocol for communication must exist. There are many such protocols which usually have implementations that include agents of some degree of sophistication. These *agent frameworks*, or *middleware* not only provide communication mechanisms but also make the programmer's work easier by including useful agent functionality that can be reused.

The Java Agent Development Framework (JADE) is a free framework that complies with the FIPA specifications for a multi-agent system, thus allowing JADE agents to interoperate with other FIPA-compliant agents. It provides support for message transport, encoding and parsing, and agent life-cycle issues. It is written completely in Java and extensive documentation and tutorials are available at the JADE home page.

JADE is not a BDI-architecture such as JACK or 3APL. The life-cycle support it provides is for agent setup, run-time and termination. This allows for any agent architecture to be implemented. For complete information about JADE, please refer to <http://jade.tilab.com/>.

3.3.2 Current uses of this technology

JADE has been around since 1997 and has been used in a variety of fields - for business, research and development, and academic uses. A general categorization of the uses of JADE are:

1. **Mobile Applications.** The LEAP add-on to JADE provides support for running JADE on mobile devices such as PDAs. This allows JADE agents to be used in Mobile Context-Aware Applications, where the agent can run on the PDA and act as a "personal agent" that helps its owner. An example of this use is the IMAGE project finished in 2002. It "integrated existing georeference services (i.e. routing, mapping, proximity search, geo-coding, GPS tracking), content providing services (lists of points of interest, e.g. restaurants, hotels, etc), introduced the personalization feature ... in order to provide the modern mobile user ... with personalized location-based global M-services". For a description of this project, see <http://www.image-project.com/>.
2. **Internet Applications.** As an implementation of the peer-to-peer architecture, JADE is ideally suited to be used over the Internet to utilise the increased connectivity and distributivity that peer-to-peer programs provide. Various applications have been considered such as e-learning, e-healthcare, and of course e-commerce/e-trading. In addition, the entertainment sector has tackled applications such as community services, multiparty gaming and content sharing. There are many such application described at <http://jade.tilab.com/>, please refer to that link for more details.
3. **Corporate Applications.** The deployment of a JADE-based system is an obvious boon for instances where information needs to be shared or tasks coordinated. In the fields of Knowledge Management and personnel administrations JADE has been proposed, and in addition

in logistics or production processes. An example of a corporate application is Swisstransplant, an agent-based system for decision making support in organ transplant centers. Refer to <http://www.swisstransplant.org> for details.

4. **Machine-to-Machine applications.** In the management of complicated systems where the number of elements and complexity of the relations are high, it is well-known that agent-based systems can markedly increase efficiency. Obvious examples are automatic control or traffic management systems, however network management is another field that is currently being studied to see if the agent paradigm can increase efficiency. JADE agents have been used in all these areas.

3.3.3 What we used this technology for

We have used JADE as a framework for a community of agents working together to provide services for users of a lablet space. JADE provided the agent communication mechanisms that we used to have agents running on multiple machines and cooperating in their tasks.

3.3.4 How successful were we?

We implemented our system using JADE without any difficulties. We implemented an extension of the BDI agent architecture ourselves which allows us to add new agents extremely easily by reusing our existing code through inheritance.

3.3.5 Other possible uses of this technology

The agent paradigm is finding a growing number of possible applications. JADE is a powerful yet flexible agent framework and so should be able to be used wherever the agent paradigm can be implemented.

An interesting paper on a simple mapping of an agent design using the Gaia methodology to the JADE framework can be found at <http://www.netobjectdays.org/pdf/02/papers/ws-ages/0815.pdf>. This is a paper entitled: "Engineering JADE agents with the Gaia Methodology". A possible area for further study may be the analysis of the above method with respect to its transferability to the ROADMAP methodology.

3.4 PDA

3.4.1 Overview

See May 7th PDA, section [2.3](#).

3.4.2 Current uses of this technology

See May 7th PDA, section [2.3](#).

3.4.3 What we used this technology for

For the final year demonstration, we used iPaq PDA to:

1. transfer image and text messages to the PDA via a wireless bluetooth connection;
2. as a guiding tool to navigate the user via text messages sent wirelessly using a bluetooth connection.

3.4.4 How successful were we?

The ability to receive image and text messages on the PDA required only minor modifications to the PDA software coded for the May 7th demonstration. This worked flawlessly, although instabilities were encountered with the Clarinox bluetooth adaptor due to a bug in the bluetooth stack (refer to Clarinox bluetooth section [3.1](#)).

Guiding comprised of the PDA receiving text only messages which worked as expected. The same program developed for receiving image and text messages were used for this purpose. Consequently, guiding was affected by the same issues of instability due to the Clarinox bluetooth stack.

3.4.5 Other possible uses of this technology

See May 7th PDA, section [2.3](#).

3.5 Recording

This section describes the technology used to record the actual demonstration of our Intelligent Lifestyle project.

3.5.1 Overview

The implementation of our agent system is built upon a JADE framework. We needed to be able to capture the video and audio stream from a Kodak USB camera that is controlled by a JADE agent and the file should be of a suitable format which can be viewed by a media player.

We found a sample program that captures the video and audio stream from a Logitech QuickCam Pro 3000 USB camera for 10 seconds and stores it on a file. It is 100% written in Java, using the media framework library (JMF) from Sun. It is a non Windows/GUI based program which shows us an example of how recording can be accomplished with a USB camera.

3.5.2 Current uses of this technology

Currently, the sample program is made available for the general public to grasp a quick understanding how to capture video from a USB camera with Java JMF. The author who wrote this sample program was interested to know if it was possible to access the USB camera from a java program. He intended to make his source code available for an educational purpose only and modifications are allowed to enable the program to work for any other USB cameras.

3.5.3 What we used this technology for

We have made modifications to the sample program so that we could capture the video and audio stream from a Kodak USB camera and the recording is scheduled by a JADE agent. After recording is finished, the file can be found at a convenient location which can be displayed by Windows Media Player. This means that the JADE agent will initiate the recording of the demonstration of our project when the actual demonstration begins and it will stop the recording when the demonstration finishes.

3.5.4 How successful were we?

We implemented the recording of our demonstration without any difficulties and the program was recorded such that time restrictions were no longer present.

3.5.5 Other possible uses of this technology

Another possible use of this recording technology would be to integrate it with a scheduler and a local webserver to publish periodically live video fragments on the Internet but this is irrelevant to the project.

3.6 Speech-to-Text

This section describes the technologies used to demonstrate speech-to-text.

3.6.1 Overview

Speech-to-text is the ability for a microphone equipped PC to convert spoken words into text that can be used by a program. Team Daedalus used Microsoft Speech SDK 5.1 to implement speech-to-text. Microsoft Speech SDK is a software development kit for building speech engines and applications for Microsoft Windows. The SDK contains the Microsoft Win32-compatible speech application programming interface (SAPI), the Microsoft continuous speech recognition engine and Microsoft concatenated speech synthesis (or text-to-speech) engine, a collection of speech-oriented development tools for compiling source code and executing commands.

3.6.2 Current uses of this technology

1. Speech recognition engines of Speech SDK 5.1 is used to recognize the speech that being dictated or recorded.
2. The built in grammar compiler interface allows developers to replace most keyboard tasks with speech commands. This can be used in playing games, accessing menu and voice-activate command by computer microphone instead of using menu options or keyboard accelerators. This technology is widely used for web-based speech applications as well.

3.6.3 What we used this technology for

We used speech-to-text technology as the main form of user input to the system. Having speech-to-text meant that the user could more naturally interact with the Intelligent Lablet system without having to use a keyboard or mouse.

3.6.4 How successful were we?

Team Daedalus successfully implemented speech-to-text using speech recognition engines and grammar compiler interfaces. The accuracy of normal speech dictation was around 60% as it doesn't dictate exact words of the speech. To solve this problem, we implemented another command dictation using grammar compiler interfaces to dictate certain commands so that we could manipulate the system when certain commands or words being dictated.

3.6.5 Other possible uses of this technology

Microsoft Speech Server 2004 which is more powerful and accurate than Microsoft Speech SDK 5.1 was launched on March 2004 which makes the business value of speech technology more broadly available to mainstream enterprise companies. Microsoft Speech Server 2004 offer a single platform that combines Web technologies, speech-processing services and telephony capabilities. This enables companies to unify their Web and telephony infrastructure and extend existing or new ASP.NET Web applications for speech-enabled access from telephones, mobile phones, Pocket PCs and Smartphones. With these tools, Microsoft has completely leveraged that power and ease of use for building speech-enabled applications, with a minimal learning curve for Visual Studio developers. Additionally, Microsoft is researching on increasing language coverage, further enhanced natural language support, speech recognition for embedded devices and telecommunication carrier

needs. This will enable future versions of Microsoft Speech Server to serve both the enterprise and telecommunication 04 customers.

3.7 Text-to-Speech

3.7.1 Overview

For text-to-speech the Microsoft Speech SDK(5.1) was used which supports C++, C # and VB. It contains many tools, information, examples and applications that are quite useful in learning the technology. The central API for text-to-speech (TTS) is ISpVoice. Using this, applications can add TTS support such as speaking text, modifying speech characteristics, changing voices, as well as responding to real-time events while speaking. We found the Speech SDK quite useful and simple to use when implementing text-to-speech.

3.7.2 Current uses of this technology

This technology is used for the verbal output of a variety of programs. The uses of this technology seem pretty standard across the board - to communicate information to the user via an audio stream. We also used the Speech SDK for our speech recognition but that won't be detailed in this section.

Our project implemented a fairly simple version of the technology as basic text-to-speech was all that was required.

3.7.3 What we used this technology for

We used this technology as the main method of communicating and interacting with the user. The was done by parsing data to the SDK in XML format.

3.7.4 How successful were we?

We were successful in implementing this technology for our project. The system allowed for changes in volume, voice and rate via methods. The only difficulties were with getting Java to communicate with the C++ code (required since all Agents were written in Java, whilst the SDK could only be accessed using C++). For this we used the JNI interface which unfortunately didn't have much support. Examples of simple applications could be found easily, but more complex ones were in short supply. In the end however, we were able to get Java to communicate with C++. The documentation of the Speech SDK was quite comprehensive and this combined with programming forums provided more than enough support to get this part of the project completed.

3.7.5 Other possible uses of this technology

No other possible uses of this technology has been identified.

3.8 Web server

This section describes the technologies used to demonstrate the Web server.

3.8.1 Overview

Team Daedalus used Java to implement the Web server and used Java Servlet API to extend the functionality of the Web server. The Java Servlet API is a Standard Extension to the Java platform that provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server. The servlet provides a component-based, platform-independent method for building Web-based applications.

3.8.2 Current uses of this technology

None has been identified.

3.8.3 What we used of this technology

For the final demonstration the Web server was required to:

1. store the location information of a particular device
2. store the message and image send from PC to PDA
3. receive an web request from the PC/PDA via HTTP connection
4. response to the request by sending requested information in XML format , which could be the location information of an particular device/the message and image for the particular device.

3.8.4 How successful were we

The framework for the Web server was provided to us by Fernando Koch of the IS department who has had much experience with mobile web-based applications. Modifications were made to his code to cater for our requirements, which we successfully completed. Requests were able to be sent to the Web server with the correct information, and the response to the web request was completed without any difficulties.

3.8.5 Other possible uses of this technology

The Web server may have the following possible uses:

1. it can used as an online database
2. it can be used as a location server to store the location information of devices

4 Change Log

Date	Section	Descriptions
-	-	-

Table 2: Change Log